A Modal Logic for Abstract Delta Modeling

Frank de Boer*
CWI, Amsterdam
Leiden University
The Netherlands
f.s.de.boer@cwi.nl

Michiel Helvensteijn*
CWI, Amsterdam
Leiden University
The Netherlands
michiel.helvensteijn@cwi.nl

Joost Winter
CWI, Amsterdam
The Netherlands
j.winter@cwi.nl

ABSTRACT

Abstract Delta Modeling is a technique for implementing (software) product lines. Deltas are put in a partial order which restricts their application and are then sequentially applied to a core product in order to form specific products in the product line. In this paper we explore the semantics of deltas in more detail. We regard them as relations between products and introduce a multi-modal logic that may be used for reasoning about their effects. Our main innovation is a modality for partially ordered sets of deltas. We prove strong completeness results on both the frame level and the model level and demonstrate the logic through an example.

1. INTRODUCTION

Delta Modeling [12, 13, 14] is designed as a technique for implementing *software product lines* [11]: a way to optimally reuse code between software products which differ only by which features they support. The code is divided into units called *deltas*, which can incrementally transform a core product in order to generate a product in the product line.

Clarke et al. [4, 5] described delta modeling in an abstract algebraic manner called the *Abstract Delta Modeling* (ADM) approach. In that work, delta modeling is not restricted to software product lines, but rather product lines of any domain. It gives a formal description of deltas, how they can be applied to *products*, how they can be combined, how they can be linked to features from the feature model, as well as how to avoid and resolve implementation conflicts. Most notably, they put deltas in a partial order to restrict their order of application. This allowed for an exact specification of dependency between deltas, as well as the implementation of desired feature interaction and the resolution of conflict with a minimum of code duplication.

At its core, ADM is about deltas that can transform one product into another product. We need a way to specify and reason about the semantics of deltas, and what effect they have on the features that are supported by a product. We need a way to specify that a delta implements a specific feature or that a delta refrains from breaking an existing feature. We need a way to prove that if certain local guarantees are met, that specific global properties, such as *product line completeness* [8, 9], are then guaranteed to hold.

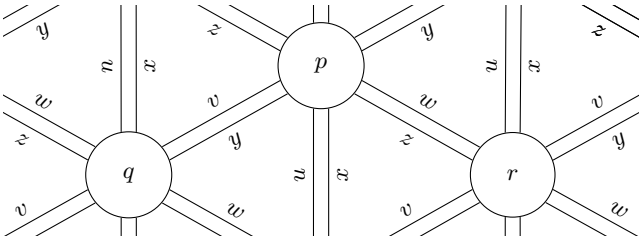

Figure 1: Example view of a delta frame with products p, q, r and deltas u, v, w, x, y, z currently visible

In this paper we introduce a modal logic in order to reason about the semantics of deltas. Basically, we take the set of all possible products as the set of *worlds* in our frame (Figure 1). We then model deltas as binary relations on this set. In previous work, all deltas were deterministic (functional). We now generalize the notion of delta, and allow them to be nondeterministic, as well as non-terminating. In our logic, we want to be able to make judgements such as

$$\models \langle d \rangle\, f \qquad\qquad \models [d]\, f$$

meaning "delta d may implement feature f" (left) and "delta d must implement feature f" (right). Or perhaps, for all ϕ:

$$\models \langle d \rangle\, \phi \to [d]\, \phi \qquad\qquad \models [d]\, \phi \to \langle d \rangle\, \phi$$

meaning "delta d is deterministic" (left) and "delta d always terminates" (right). Note that we implicitly quantify over all products that the delta may be applied to.

We also introduce an additional modalities, representing *delta models* (partially ordered sets of deltas, Definition 4), in order to make judgements such as

$$\models [DM]\, (f \wedge g \wedge h)$$

meaning "delta model DM implements features f, g and h".

The paper is structured as follows. Sections 2 and 3 summarize the relevant theory of abstract delta modeling and modal logic respectively. Section 4 introduces both the syntax and semantics of our modal logic on a frame level. It also proves strong completeness. Then, Section 5 introduces proposition letters and explores our logic on a model level. Section 6 concludes and discusses related and future work.

*These authors are partially supported by the EU project FP7-231620 HATS: Highly Adaptable and Trustworthy Software using Formal Models (http://www.hats-project.eu)

2. ABSTRACT DELTA MODELING

To make this paper self-contained, we now repeat the relevant theory from ADM. For more detailed information, we refer the reader to [4, 5]. Readers familiar with the theory can skip this section.

2.1 Products and Deltas

First, we assume a set of *products*, $\mathcal{P}$. The set of possible modifications to products forms a delta monoid, as follows:

DEFINITION 1 (DELTA MONOID). *A* delta monoid *is a monoid $(\mathcal{D}, \cdot, \epsilon)$, where $\mathcal{D}$ is a set of product modifications (referred to as* deltas*), and the operation $\cdot : \mathcal{D} \times \mathcal{D} \to \mathcal{D}$ corresponds to their sequential composition. $y \cdot x$ denotes the modification applying first x and then y. The neutral element ϵ of the monoid corresponds to modifying nothing.*

Applying a delta to a product results in another product. This is captured by the notion of delta action. The following definition differs from previous work [4, 5], in which deltas were always deterministic, and would always terminate. The notion of nondeterministic delta action allows for both nondeterminism and nontermination, by resulting in a set of products, rather than a single product.

DEFINITION 2 (NONDETERMINISTIC DELTA ACTION). *A* nondeterministic delta action *is an operation $-(-) : \mathcal{D} \times \mathcal{P} \to \mathscr{P}(\mathcal{P})$. If $d \in \mathcal{D}$ and $p \in \mathcal{P}$, then $d(p) \subseteq \mathcal{P}$ is the set of products that may result from applying delta d to product p. It satisfies the conditions $(y \cdot x)(p) = \bigcup_{q \in x(p)} y(q)$ and $\epsilon(p) = \{p\}$.*

This all leads to the notion of a *deltoid*, which describes all building blocks necessary to create a product line in a concrete domain.

DEFINITION 3 (DELTOID). *A* deltoid *is a quintuple $(\mathcal{P}, \mathcal{D}, \cdot, \epsilon, -(-))$, where $\mathcal{P}$ is a product set, $(\mathcal{D}, \cdot, \epsilon)$ is a delta monoid and $-(-)$ is a nondeterministic delta action operator.*

A delta model describes the set of deltas required to build a specific product, along with a strict partial order on those deltas, restricting the order in which they may be applied.

DEFINITION 4 (DELTA MODEL). *A* delta model *is a pair $(D, \prec)$, where $D \subseteq \mathcal{D}$ is a finite set of deltas and $\prec \subseteq D \times D$ is a strict partial order on D. $x \prec y$ states that x must be applied before y, though not necessarily directly before.*

The partial order represents the intuition that a delta applied later has full access to earlier deltas and more authority over modifications to the product.

The semantics of a delta model is defined by its derivations. A *derivation* is a delta formed by a sequential composition of all deltas from D, in some linearization of the partial order.

DEFINITION 5 (DERIVATION). *Given a delta model $DM = (D, \prec)$, its* derivations *are defined to be*

$$derv(DM) \stackrel{\text{def}}{=} \left\{ x_n \cdot \ldots \cdot x_1 \;\middle|\; \begin{array}{l} x_1, \ldots, x_n \text{ is a linear} \\ \text{extension of } (D, \prec) \end{array} \right\}.$$

Observe that when D is empty, $derv(DM) = \{\epsilon\}$. Also note that $derv(DM)$ may potentially generate more than one distinct derivation, as non-commutative deltas may be applied in different orders. Techniques for ensuring a unique derivation (and thus a unique product) may be found in [4, 5]. In this paper, we make no assumptions about the determinism of basic deltas or the unambiguity of delta models.

For specific product lines, a set $\mathcal{F}$ of relevant feature labels is introduced. Eventually deltas are linked to feature labels, so we can generate a delta model for each legal combination of features. We do not describe this process in detail here, but we do repeat the concept of feature model, since it will be referenced later:

DEFINITION 6 (FEATURE MODEL). *A* feature model *$\Phi \subseteq \mathscr{P}(\mathcal{F})$ is a set of sets of feature labels from $\mathcal{F}$. Each $F \in \Phi$ is a set of feature labels corresponding to a valid* feature configuration*, i.e. a set of features that may be active together.*

3. MODAL LOGIC

In this section, we recall a number of essential notions from the theory of modal logic [2]. We define the basic language, its semantics and the syntactic notion of a proof in general terms. Following this, in the next section, we will instantiate this theory with a language in which the modalities correspond to the deltas from our underlying abstract delta modeling framework.

3.1 Language and Semantics

We will be concerned with a basic multi-modal language in which we have a set of proposition letters, and a set of labeled modalities. In order to keep the story simple and accessible, we will only concern ourselves with unary modalities, as well as, in Section 5, nullary modalities which can be regarded as playing the role of propositional constants. In principle, however, modalities can have any arity. This basic modal language consists of the following terms:

$$\phi \;::=\; \bot \mid p \mid \phi \vee \phi \mid \neg\phi \mid \textcircled{p} \mid \langle d \rangle\, \phi$$

Here, $\langle d \rangle$ is any unary modality labeled with d, $\textcircled{p}$ is any nullary modality labeled with p and p is any proposition letter taken from a set Ξ of proposition letters.

A frame $\mathfrak{F}$ over this language consists of a set W of worlds and, for each nullary modality $\textcircled{p}$ a predicate $U_p \subseteq W$ and for each unary modality $\langle d \rangle$, a binary relation $R_d \subseteq W \times W$.

A model $\mathfrak{M}$ over a frame consists of a frame and a valuation function $V : \Xi \to \mathscr{P}(W)$, mapping proposition letters to sets of worlds. We can now, given a model $\mathfrak{M}$ and world $w \in W$, define the modal satisfaction relation $\vDash$ as follows:

$\mathfrak{M}, w \vDash \bot$	never	
$\mathfrak{M}, w \vDash p$	iff	$w \in V(p)$
$\mathfrak{M}, w \vDash \phi \vee \psi$	iff	$\mathfrak{M}, w \vDash \phi$, or $\mathfrak{M}, w \vDash \psi$
$\mathfrak{M}, w \vDash \neg\phi$	iff	not $\mathfrak{M}, w \vDash \phi$
$\mathfrak{M}, w \vDash \textcircled{p}$	iff	$w \in U_p$
$\mathfrak{M}, w \vDash \langle d \rangle\, \phi$	iff	there exists a $v \in W$ with $(v, w) \in R_d$ and $\mathfrak{M}, v \vDash \phi$

We regard $\phi \wedge \psi$, $\phi \to \psi$ and $[d]\,\phi$ as abbreviations for $\neg(\neg\phi \vee \neg\psi)$, $\neg\phi \vee \psi$ and $\neg\langle d \rangle \neg\phi$, respectively.

We furthermore write $\mathfrak{M} \vDash \phi$ and say that ϕ is *globally true in* $\mathfrak{M}$ iff for all worlds w, we have $\mathfrak{M}, w \vDash \phi$. Given a frame $\mathfrak{F}$, we write $\mathfrak{F}, w \vDash \phi$ and say ϕ is valid at world

w iff for all models $\mathfrak{M}$ based on $\mathfrak{F}$, we have $\mathfrak{M}, w \vDash \phi$. We furthermore write $\mathfrak{F} \vDash \phi$ and say ϕ is valid on $\mathfrak{F}$ iff for all worlds w, we have $\mathfrak{F}, w \vDash \phi$. When we want to restrict the semantic entailment to a certain class of structures S, we superscribe $\vDash$ with S, as in $\vDash_{\mathsf{S}}$.

Given a set of formulas Γ and a class of structures S (either models or frames), we say that ϕ is a *local consequence* of Γ, and write $\Gamma \vDash_{\mathsf{S}} \phi$, iff, for all models $\mathfrak{M}$ (possibly based on frames) from S, and all worlds $w \in W$:

$$\mathfrak{M}, w \vDash \phi \qquad \text{whenever} \qquad \mathfrak{M}, w \vDash \Gamma.$$

Likewise, given a set of formulas Γ and a class of structures S, we say ϕ is a *global consequence* of Γ, and write $\Gamma \vDash^{g}_{\mathsf{S}} \phi$, iff, for all models $\mathfrak{M}$ from S, we have

$$\mathfrak{M} \vDash \phi \qquad \text{whenever} \qquad \mathfrak{M} \vDash \Gamma.$$

3.2 Proof Theory

Definition 7 (Normal Modal Logic). *Given any modal language, a* normal modal logic *is a set of formulas Λ containing all propositional tautologies, the formula* **K***:*

$$[d]\,(p \to q) \to ([d]\,p \to [d]\,q),$$

the formula **Dual***:*

$$\langle d \rangle\, p \leftrightarrow \neg\, [d]\, \neg p$$

(for all modalities d) and closed under:

- Modus ponens*: if $\phi \in \Lambda$ and $\phi \to \psi \in \Lambda$, then $\psi \in \Lambda$;*
- Uniform substitution*: if $\phi \in \Lambda$, then $\phi[\psi/p] \in \Lambda$ for all proposition letters p and formulas ϕ; and*
- Generalization*: if $\phi \in \Lambda$, then $[d]\,\phi \in \Lambda$ for all modalities d.*

Given any set of formulas Γ, a smallest normal modal logic containing all formulas in Γ always exists, and will be called the normal modal logic *generated by* Γ.

Given a normal modal logic Λ, we write

$$\vdash_{\Lambda} \phi$$

to denote $\phi \in \Lambda$, and

$$\Gamma \vdash_{\Lambda} \phi$$

to express that there are formulas $\psi_1, \ldots, \psi_n$ such that

$$\vdash_{\Lambda} \Big(\bigwedge_{1 \le i \le n} \psi_i \Big) \to \phi.$$

Alternatively, we can also regard the relation $\vdash$ in terms of a proof system. Here, we regard **K** and **Dual**, together with all propositional tautologies as axioms, and regard the earlier closure properties (modus ponens, uniform substitution, and generalization) as proof rules.

A normal modal logic Λ is called *strongly complete* with respect to a class S of frames, if, when for any set of formulas Γ and any formula ϕ, $\Gamma \vDash_{S} \phi$ implies $\Gamma \vdash_{\Lambda} \phi$. The normal modal logic **K**, generated by the empty set, is strongly complete with respect to the class of all frames [2].

4. DELTA FRAMES

One of the primary goals of this paper is to reason about abstract delta modeling using the language and techniques of modal logic. A good starting point, before moving on to an axiomatic characterization (in which we are concerned with issues such as completeness), is to describe delta modeling using Kripke frames.

4.1 Relational Deltas

For the convenience of the formalism described in the remainder of the paper, we now start working in a more concrete deltoid, in which deltas are relations between products.

Definition 8 (Relational Deltoid). *A* relational deltoid $(\mathcal{P}, \mathcal{D}, \cdot, \epsilon, -(-))$ *is a deltoid in which $\mathcal{D} = \mathscr{P}(\mathcal{P} \times \mathcal{P})$.*

For a complete charactarization of the deltoid and a solid link to earlier work [4, 5], we also need to define delta action (Definition 2) concretely, but this is quite straightforward.

Definition 9 (Relational Delta Action). *A* relational delta action *is an operation $-(-) : \mathcal{D} \times \mathcal{P} \to \mathscr{P}(\mathcal{P})$ such that for all $d \in \mathcal{D}$ and all $p \in \mathcal{P}$:*

$$d(p) \stackrel{\text{def}}{=} \{\, q \in \mathcal{P} \mid (p, q) \in d \,\}$$

This implicitly defines sequential composition $\cdot$ as relation composition and the empty delta ϵ as the identity relation.

The paper loses no generality with this approach. The only real difference is that there can no longer exist multiple distinct deltas that represent the same relation.

4.2 Delta Terms

We define the set of *delta terms* (which can be seen as the syntactic counterparts of deltas) as the smallest set such that:

1. Every delta has a corresponding basic delta term d,
2. Given delta terms d_1 and d_2, $d_2 \cdot d_1$ and $d_1 \cup d_2$ are delta terms, and
3. Given a finite set D of delta terms, and a partial order $\prec : D \times D$, $(D, \prec)$ is a delta term.

From here onward, we use the set of delta terms to label our set of unary modalities. i.e. for each delta term d, there exist unary modalities $\langle d \rangle$ and $[d]$. We are not using nullary modalities yet, but they become useful in Section 5.2.

4.3 Frames and Relations

A concrete relational deltoid uniquely defines a delta frame $\mathfrak{F} = (W, R_{d_1}, \ldots)$. The set of worlds W is the set of products $\mathcal{P}$ and the set of binary relations R_{d_i} is the set of deltas $\mathcal{D}$.

Definition 10. *The relation R_d is the delta corresponding to basic delta term d. We define the binary relations corresponding to compound delta terms inductively, in terms of basic delta terms. First, union and composition:*

$$\begin{aligned} R_{d_1 \cup d_2} &\stackrel{\text{def}}{=} R_{d_1} \cup R_{d_2} \\ R_{d_2 \cdot d_1} &\stackrel{\text{def}}{=} \{(p_3, p_1) | \exists p_2 ((p_3, p_2) \in R_{d_2} \wedge (p_2, p_1) \in R_{d_1})\} \end{aligned}$$

Finally, the binary relation corresponding to a partial order $(D, \prec)$ on delta terms can be described in terms of derivations of this partial order as follows:

$$R_{(D,\prec)} \stackrel{\text{def}}{=} \bigcup_{d \in derv((D,\prec))} R_d,$$

Using *derv* (Definition 5) here is a bit of an abuse of notation, as it is defined on deltas, not delta terms. However, a delta term version can be defined analogously. Note that if the relations corresponding to the delta terms in D are deterministic (functional) and the partial order $(D, \prec)$ has a unique derivation, the relation $R_{(D,\prec)}$ is deterministic as well. Note also that we can characterize composition in terms of partial orders:

$$R_{d_2 \cdot d_1} = R_{(\{d_1,d_2\},\{(d_1,d_2)\})}$$

and, conversely, we can characterize partial orders in terms of union and composition.

DEFINITION 11 (DELTA FRAMES). *Let $\Delta\mathsf{F}$, the class of Delta frames, be the class of all frames, with a underlying set of delta terms as modalities, satisfying the relational equalities from Definition 10.*

We now introduce the following useful notation:

NOTATION 12. *For a given partially ordered set $DM = (D, \prec)$ and subset $D' \subseteq D$, we define the notation:*

$$DM \setminus D' \stackrel{\text{def}}{=} (D \setminus D', \prec')$$

where $\prec'$ is $\prec$ restricted to $D \setminus D'$.

From Definition 10, the following proposition follows straightforwardly:

THEOREM 13. *Given a nonempty delta model $DM = (D, \prec)$ and any formula ϕ, we have*

$$\vDash_{\Delta\mathsf{F}} \langle DM \rangle \phi \leftrightarrow \bigvee_{d \text{ minimal}} \langle d \rangle \langle DM \setminus \{d\} \rangle \phi$$

and for the empty delta model $(\varnothing, \varnothing)$, we have

$$\vDash_{\Delta\mathsf{F}} \langle (\varnothing, \varnothing) \rangle \phi \leftrightarrow \phi$$

PROOF. Induction on the size of D. □

It is worthwhile to note that the above theorem is similar to what is known as the *expansion law* of the process algebra CCS [10]. Because delta models are finite and do not contain cycles in our case the expansion law in combination with other axioms allows a complete reduction to basic delta terms, as explained in more detail below.

Dually, the semantic entailment

$$\vDash_{\Delta\mathsf{F}} [DM] \phi \leftrightarrow \bigwedge_{d \text{ minimal}} [d] [DM \setminus \{d\}] \phi$$

is also valid as a direct consequence of Theorem 1.

In the next section we will discover that the normal modal logic generated by these formulas (together with axioms for union and composition) is strongly complete with respect to the class of delta frames.

4.4 Completeness

DEFINITION 14. *Define the modal logic $\mathbf{K\Delta}$ as the smallest normal modal logic containing all instances of the following axiom schemata:*

1. $\langle DM \rangle \phi \leftrightarrow \bigvee_{d \text{ min.}} \langle d \rangle \langle DM \setminus \{d\} \rangle \phi$ *(nonempty DM)*
2. $\langle (\varnothing, \varnothing) \rangle \phi \leftrightarrow \phi$*;*
3. $\langle d_2 \cdot d_1 \rangle \phi \leftrightarrow \langle d_1 \rangle \langle d_2 \rangle \phi$*; and*
4. $\langle d_1 \cup d_2 \rangle \phi \leftrightarrow (\langle d_1 \rangle \phi \vee \langle d_2 \rangle \phi)$.

We call instantiations of these axiom schemata '$\mathbf{\Delta}$ axioms'. These allows us to formulate the following completeness result, after defining a translation function t as follows (that t is well-defined trivially follows from defining a fitting complexity function on formulas):

DEFINITION 15.

$$\begin{array}{lcl}
t(f) & \stackrel{\text{def}}{=} & f \quad \textit{for proposition letters } f \\
t(\neg\phi) & \stackrel{\text{def}}{=} & \neg t(\phi) \\
t(\phi \vee \psi) & \stackrel{\text{def}}{=} & t(\phi) \vee t(\psi) \\
t(\langle d \rangle \phi) & \stackrel{\text{def}}{=} & \langle d \rangle t(\phi) \quad \textit{for basic delta terms } d \\
t(\langle d_2 \cdot d_1 \rangle \phi) & \stackrel{\text{def}}{=} & t(\langle d_1 \rangle \langle d_2 \rangle \phi) \\
t(\langle DM \rangle \phi) & \stackrel{\text{def}}{=} & \bigvee_{d \in derv(DM)} t(\langle d \rangle \phi)
\end{array}$$

The idea behind this function is to translate any formula into an equivalent formula in which all unary modalities are labeled only by basic delta terms. This enables us to forget about compound delta terms, allowing us to construct our completeness proof in terms of the completeness of $\mathbf{K}$ w.r.t. the class of all frames.

LEMMA 16. *For all Γ and ϕ, we have:*

1. $\Gamma \vDash_{\Delta\mathsf{F}} \phi$ *iff* $\Gamma \vDash_{\Delta\mathsf{F}} t(\phi)$*;*
2. $\Gamma \vdash_{\mathbf{K\Delta}} \phi$ *iff* $\Gamma \vdash_{\mathbf{K\Delta}} t(\phi)$*; and*
3. $\Gamma \vDash_{\Delta\mathsf{F}} t(\phi)$ *iff* $\Gamma \vDash t(\phi)$

PROOF. The first and second part of the lemma can be proven by induction (on the complexity of formulas as well as that of delta terms); the third part follows from the observation that for any translated formula, only the relations corresponding to basic delta terms are used: hence, we are simply treating our delta frame as a regular frame. □

THEOREM 17. $\mathbf{K\Delta}$ *is strongly complete w.r.t. the class of delta frames.*

PROOF. This amounts to saying that, for any Γ and ϕ, if $\Gamma \vDash_{\Delta\mathsf{F}} \phi$, then $\Gamma \vdash_{\mathbf{K\Delta}} \phi$. But, if $\Gamma \vDash_{\Delta\mathsf{F}} \phi$, then, by the first part of Lemma 1, we have $\Gamma \vDash_{\Delta\mathsf{F}} t(\phi)$, and by part 3 of Lemma 1, we now have $\Gamma \vDash t(\phi)$. Completeness of $\mathbf{K}$ now gives $\Gamma \vDash_{\mathbf{K}} t(\phi)$, and because $\mathbf{K} \subseteq \mathbf{K\Delta}$, we also get $\Gamma \vDash_{\mathbf{K\Delta}} t(\phi)$. Finally, part 2 of Lemma 1 now yields $\Gamma \vDash_{\mathbf{K\Delta}} \phi$). □

5. MODELS ON DELTA FRAMES

As we can now reason on the frame level with the proof system of Section 4, we would also like to reason on the model level.

Recall that a model $\mathfrak{M} = (\mathfrak{F}, V)$ is a frame augmented with a valuation function which maps proposition letters from Ξ to the set of worlds in which they are true. Our worlds are products from $\mathcal{P}$. What we want to reason about is the *features* that are implemented by those products, so we propose that $\mathcal{F} \subseteq \Xi$. We would like to prove properties about the effect of deltas on specific features given axiomatic characterizations of specific models.

5.1 Semantic Feature Model

In Definition 6 we see features as labels. A feature model Φ indicates which features are allowed to be selected together on a conceptual level. However, if we have $\mathfrak{M}, w \vDash_{\Delta\mathsf{F}} f$ for some $f \in \mathcal{F}$, it means that feature f is actually implemented in product w. It is a semantic judgment.

An interesting relation exists however. A (syntactic) feature model is only sensible if all of its feature configurations can actually be implemented. We define a semantic feature model as follows:

Definition 18 (Semantic Feature Model). *Given a model $\mathfrak{M}$, we define its semantic feature model $\Phi_{\mathfrak{M}} \subseteq \mathscr{P}(\mathcal{F})$ as the set of sets of features that can semantically be implemented together:*

$$\Phi_{\mathfrak{M}} \stackrel{\text{def}}{=} \{\, V'(w) \cap \mathcal{F} \mid w \in W \,\}$$

where $V' : W \to \mathscr{P}(\Xi)$ is the function mapping each world to the set of proposition letters that are true there:

$$V'(w) \stackrel{\text{def}}{=} \{\, p \in \Xi \mid w \in V(p) \,\}$$

We expect a sensible syntactic feature model to be a subset of the semantic feature model:

$$\Phi \subseteq \Phi_{\mathfrak{M}}$$

meaning that all valid feature configurations contain only features that can potentially be implemented together.

5.2 Proof System

Note that the proof system from Section 4 is not sound with respect to global semantic entailment on models. For example, consider the following 'proof':

(1)	$f \to [d]\, g$	axiom
(2)	$f \to [d] \neg g$	uniform substitution on g

So we have

$$f \to [d]\, g \;\vdash_{\mathbf{K}}\; f \to [d] \neg g,$$

but at the same time the (global) semantic consequence

$$f \to [d]\, g \;\vDash^{g}\; f \to [d] \neg g$$

is easily seen to be false. The culprit here is our usage of uniform substitution. This proof rule produces new validities from old validities, but it does not preserve truth on a model level. We still need the uniform substitution rule, however, to prove truths such as:

(1)	$p \vee \neg p$	propositional tautology
(2)	$[d]\, f \vee \neg\, [d]\, f$	uniform substitution on p

The trick is to allow uniform substitution only on newly produced proposition-letters, but not on the original features in our axioms. This may be accomplished by first transforming all feature propositions in our axioms to nullary modalities, on which uniform substitution does not apply. We can then prove any valid formula in the proof system of frames. We first define the following translation:

Definition 19.

$$\begin{array}{lcl} u(f) & \stackrel{\text{def}}{=} & \textcircled{f} \quad \textit{for proposition letters } f \\ u(\neg\phi) & \stackrel{\text{def}}{=} & \neg\, u(\phi) \\ \vdots & & \end{array}$$

For the other shapes of formulas the u translation is simply propagated down to the proposition letters, leaving everything else unchanged. We also lift the function u to sets of formulas in the expected manner.

Furthermore, we also define a translation function (overloading the earlier name u) from models to frames, dropping the valuation function V but augmenting the frame with, for every proposition letter in the model, a unary relation (representing a nullary modality) which holds at precisely the worlds in which this proposition letter was true in V. This enables us to formulate the following translation lemma:

Lemma 20. *For all models $\mathfrak{M}$, worlds w and formulas ϕ:*

$$\mathfrak{M}, w \vDash \phi \qquad \textit{iff} \qquad u(\mathfrak{M}), w \vDash u(\phi)$$

and

$$\mathfrak{M} \vDash \phi \qquad \textit{iff} \qquad u(\mathfrak{M}) \vDash u(\phi).$$

Proof. Induction on the complexity of formulas. The basic propositional case trivially follows from our construction of nullary modalities in terms of propositional letters. □

This lemma enables us to prove the following soundness result w.r.t. global truth on the model level:

Theorem 21. *For all sets of formulas Γ and all formulas ϕ:*

$$\textit{if } u(\Gamma) \vdash_{\mathbf{K\Delta}} u(\phi), \textit{ then } \Gamma \vDash^{g}_{\Delta\mathsf{F}} \phi$$

Proof. Assume $u(\Gamma) \vdash_{\mathbf{K\Delta}} u(\phi)$. Let $\mathfrak{M}$ be a model (based on a delta frame) such that $\mathfrak{M} \vDash \Gamma$. Then, by Lemma 2, we have $u(\mathfrak{M}) \vDash u(\Gamma)$. Now let Λ be the logic of the class of delta frames $\{\mathfrak{F} \in \Delta\mathsf{F} \mid \mathfrak{F} \vDash u(\Gamma)\}$. Because Λ is a normal modal logic, it is closed under proof rules, and hence it follows from the fact that $u(\Gamma) \subseteq \Lambda$ that $u(\phi) \in \Lambda$. It follows that $u(\phi)$ is valid on this class of frames, so we have $u(\mathfrak{M}) \vDash u(\phi)$, and by another application of Lemma 2, we get $\mathfrak{M} \vDash \phi$. Hence, $\Gamma \vDash^{g}_{\Delta\mathsf{F}} \phi$. □

5.3 Relative Completeness

In Hoare logics *relative* completeness has been established for classes of models which allow the expressibility in the logic of weakest preconditions [1]. For example in [7] a class of arithmetic models has been introduced which allow expressibility in the logic of weakest preconditions by means of arithmetically based encoding techniques. Following this general approach to completeness of Hoare logics we want to identify a class of models for which the converse of the above proposition 3 holds. More specifically, we want to identify a set of models $\mathfrak{M}$ for which there exists an axiomatisation $\Gamma_{\mathfrak{M}}$ in $\mathbf{K\Delta}$ such that $\mathfrak{M} \vDash \phi$ implies $u(\Gamma_{\mathfrak{M}}) \vdash_{\mathbf{K}} u(\phi)$. Note that in our modal logic $\mathbf{K\Delta}$ weakest preconditions of a delta d and postcondition ϕ can be directly expressed by formulas of the form $[d]\,\phi$. A natural set of models to consider are those models which allow the expression of such weakest preconditions in terms of a logical combination of features themselves.

Definition 22 (Precondition Expressibility). *A model $\mathfrak{M}$ allows the expression in $\mathbf{K\Delta}$ of weakest preconditions iff for every formula $[d]\,\phi$, where d is a basic delta term and ϕ is a boolean combination of features, there exists a boolean combination of features ϕ' such that*

$$\mathfrak{M}, w \vDash [d]\,\phi \qquad \textit{iff} \qquad \mathfrak{M}, w \vDash \phi'$$

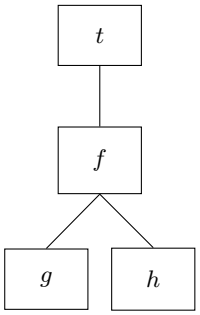

Figure 2: Example feature model

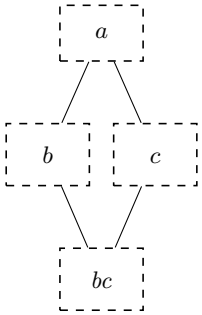

Figure 3: Example delta model *DM*

For any model $\mathfrak{M}$ let $\Gamma_{\mathfrak{M}}$ denote the propositional theory of its underlying semantic feature model extended with the theory $\mathbf{WP}(\mathfrak{M})$ defined by

$$\{\, [d]\,\phi \leftrightarrow \phi' \mid \mathfrak{M} \vDash [d]\,\phi \leftrightarrow \phi' \,\}$$

where d is a basic delta term, and both ϕ and ϕ' are boolean combinations of features. We have the following relative completeness theorem.

THEOREM 23. *For any model $\mathfrak{M}$ that allows the expression in $\mathbf{K\Delta}$ of weakest preconditions we have*

$$\text{if } \mathfrak{M} \vDash \phi, \text{ then } u(\Gamma_{\mathfrak{M}}) \vdash_{\mathbf{K\Delta}} u(\phi)$$

for every formula ϕ.

PROOF. It is suffices to show that using the transformation function t from Definition 13 the propositional theory $\Gamma_{\mathfrak{M}}$ allows to reduce every formula ϕ to a logical combination of features. The proof proceeds by a straightforward induction on ϕ. □

Note that for models that allow the expression of weakest preconditions our modal logic $\mathbf{K\Delta}$ is in fact a *conservative extension* of the propositional logic of the underlying semantic feature models. Of particular interest is also that for deterministic delta models we only need to require that every formula $[d]\,f$, where d is a basic delta term and f is a single feature, can be expressed by a boolean combination of fatures itself.

5.4 Example

We now illustrate the use of $\mathbf{K\Delta}$ through an example proof. Say we have the feature model as shown in Figure 2. The features f, g and h are implemented by the delta model DM in Figure 3. The feature t is satisfied in some empty core product, on which we'd like to apply those deltas.

We now introduce a set of basic axioms valid in this model:

AXIOM 24 (DELTA MODEL AXIOMS).

(1)	$f \to t$	(6)	$t \to [a]\,f$
(2)	$g \to f$	(7)	$f \to [b]\,g$
(3)	$h \to f$	(8)	$f \to [c]\,h$
(4)	$g \to [c]\,g$	(9)	$g \to [bc]\,g$
(5)	$h \to [b]\,h$	(10)	$h \to [bc]\,h$

Axioms 1, 2 and 3 are due to the feature model shown in Figure 2. It is generally the case that when a subfeature is implemented its superfeature is implemented as well. Axioms 4 and 5 are due to a property we assume the underlying deltoid to have, called non-interference [8], which states that commuting deltas cannot interfere with each others features. Axioms 6 to 10 are by design of the deltas a, b, c and bc. We assume that they were developed such that a, b and c implement the features f, g and h (Axioms 6, 7 and 8), taking into account only the deltas 'above' them, and that conflict resolving delta bc [4, 5] doesn't break the features implemented by the previous deltas (Axioms 9 and 10).

Now say we have a core product $c \in \mathcal{P}$ with $c \vDash t$. We'd like to prove the following property:

PROPOSITION 25. $c \vDash [DM]\,(t \wedge f \wedge g \wedge h)$

In order to prove this property more succinctly, we introduce the following auxiliary proof rules:

LEMMA 26 (L3). *For all formulas ϕ, ψ and χ, and for all deltas $d_1, \ldots, d_n$, we have:*

$$\phi \to [d_1] \cdots [d_n]\,\psi, \quad \psi \to \chi \quad \vdash \quad \phi \to [d_1] \cdots [d_n]\,\chi$$

PROOF. By induction on n. □

LEMMA 27 (L4). *For all formulas ϕ and ψ and all deltas d, we have:*

$$\vdash \quad ([d]\,\phi \wedge [d]\,\psi) \;\leftrightarrow\; [d]\,(\phi \wedge \psi)$$

PROOF. See [2, Example 1.40]. □

PROOF OF PROPOSITION 1.

(11)	$\textcircled{t} \to [a]\,[b]\,\textcircled{g}$	l3: 6, 7
(12)	$\textcircled{t} \to [a]\,[b]\,(\textcircled{f} \wedge \textcircled{g})$	l3: 11, 2
(13)	$\textcircled{t} \to [a]\,[b]\,(\textcircled{f} \wedge \textcircled{g} \wedge [c]\,\textcircled{h})$	l3: 12, 8
(14)	$\textcircled{t} \to [a]\,[b]\,(\textcircled{g} \wedge [c]\,\textcircled{h})$	l3: 13, 2
(15)	$\textcircled{t} \to [a]\,[b]\,([c]\,\textcircled{g} \wedge [c]\,\textcircled{h})$	l3: 14, 4
(16)	$\textcircled{t} \to [a]\,[b]\,[c]\,(\textcircled{g} \wedge \textcircled{h})$	l4: 15
(17)	$\textcircled{t} \to [a]\,[b]\,[c]\,([bc]\,\textcircled{g} \wedge \textcircled{h})$	l3: 16, 9
(18)	$\textcircled{t} \to [a]\,[b]\,[c]\,([bc]\,\textcircled{g} \wedge [bc]\,\textcircled{h})$	l3: 17, 10
(19)	$\textcircled{t} \to [a]\,[b]\,[c]\,[bc]\,(\textcircled{g} \wedge \textcircled{h})$	l4: 18
(20)	$\textcircled{t} \to [a]\,[b]\,[c]\,[bc]\,(\textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	l3: 19, 2
(21)	$\textcircled{t} \to [a]\,[b]\,[c]\,[bc]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	l3: 20, 1
(22)	$\textcircled{t} \to [a]\,[b]\,[c]\,[DM_1]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	l3: 21, $\mathbf{\Delta}$
(23)	$\textcircled{t} \to [a]\,[b]\,[DM_2]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	l3: 22, $\mathbf{\Delta}$

Formula (24) is derived in a symmetric manner to (23).

(24)	$\textcircled{t} \to [a]\,[c]\,[DM_3]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	symmetric
(25)	$\textcircled{t} \to [a]\,[b]\,[DM_2]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$ $\wedge\, [a]\,[c]\,[DM_3]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	$\mathrm{I}_\wedge$: 23, 24
(26)	$\textcircled{t} \to [a]\,([b]\,[DM_2]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$ $\wedge\, [c]\,[DM_3]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h}))$	l4: 25
(27)	$\textcircled{t} \to [a]\,[DM_4]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	l3: 26, $\mathbf{\Delta}$
(28)	$\textcircled{t} \to [DM]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	l3: 27, $\mathbf{\Delta}$

where

$$\begin{aligned} DM_1 &= DM \setminus \{a, b, c\} \\ DM_2 &= DM \setminus \{a, b\} \\ DM_3 &= DM \setminus \{a, c\} \\ DM_4 &= DM \setminus \{a\} \end{aligned}$$

Then, with $c \vDash \textcircled{t}$, we have our result. □

We have skipped many steps in this proof, mostly those concerned with invoking propositional tautologies and applying modus ponens. We have kept only the most interesting steps – those that directly use our axioms.

5.5 Alternate Propositions

In this section we have chosen the set of features $\mathcal{F}$ as the significant set of propositions. But there are several reasons for choosing an alternate or additional set of propositions.

First, there may be some desired interaction between features that would not be satisfied by an implementation of any strict subset of those features. In that case, we'd want to have sets of features $\mathscr{P}(\mathcal{F}) \subseteq \Xi$ rather than individual features. We would then assume the additional axiom:

$$\vDash F \cup G \implies \vDash F \wedge \vDash G$$

for some $F, G \subseteq \mathcal{F}$. This approach was taken in [8].

Furthermore, it is possible that different products may implement the exact same features. So we may want additional proposition letters to distinguish between them in our logic and reason on a somewhat lower level. Such proposition letters may include the presence of specific classes or methods in an object oriented setting.

6. CONCLUSION

In this paper we provided a method that will be useful for further research into abstract delta modeling. The modal logic $\mathbf{K\Delta}$ forms the first lo allows us to reason more easily about the semantics of deltas and delta models in a way consistent with previous work. We prove strong completeness of the logic with respect to the class of all delta frames. We also discuss a proof technique on the level of models, prove its completeness and illustrate it through example.

The delta theory in this paper is based on Abstract Delta Modeling [4, 5]. We remain in a similarly abstract setting, yet generalize even further by removing the assumption that deltas are deterministic and terminating entities.

The logic and proof techniques in this paper will be useful for proving properties of the Delta Modeling Workflow [8, 9]. That was, in fact, partial motivation for the research in this paper.

Completeness proofs in modal logic have a long-standing history, closely tied to the history of relational semantics based on Kripke frames. A comprehensive survey of this history can be found in e.g. [2, Section 1.8].

The modal logic presented in this paper has a flavour very reminiscent of dynamic logics such as PDL [6]. A crucial difference, however, is that the logic presented here is simpler (and hence, easier to work with) due to the absence of operations such as iteration or tests. Due to this simplicity, we can easily unravel complex modalities into simpler ones, and under certain conditions even reduce them to propositional formulas, enabling us to obtain the main results from Section 5.

Possible future work following up the initial research in this paper may include work on characterizations of modal expressivity of basic properties of delta models and interactions between deltas, including positive as well as limitative results. In the case of limitative results, it may be worthwhile to look into the additional expressivity that the modal μ-calculus has to offer [3]. This additional expressivity may, for example, be required to express the condition that a conflict between two deltas is resolved by a third delta.

Another interesting research direction is the use of our logical framework in the synthesis of delta models using model checking techniques.

7. REFERENCES

[1] Jan A. Bergstra and J. V. Tucker. Expressiveness and the completeness of Hoare's logic. *J. Comput. Syst. Sci.*, 25(3):267–284, 1982.

[2] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, 2001.

[3] Julian Bradfield and Colin Stirling. *Modal mu-Calculi*, pages 721–756. Elsevier Science Inc., 2001.

[4] D. Clarke, M. Helvensteijn, and I. Schaefer. Abstract delta modeling. In *Proc. of GPCE*, pages 13–22. ACM, 2010.

[5] D. Clarke, M. Helvensteijn, and I. Schaefer. Abstract delta modeling. *Accepted to MSCS special issue*, 2012.

[6] M. J. Fischer and R. E. Landner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.

[7] David Harel. Arithmetical completeness in logics of programs. In *ICALP*, pages 268–288, 1978.

[8] M. Helvensteijn. Delta Modeling Workflow. In *Proceedings of the 6th International Workshop on Variability Modelling of Software-intensive Systems, Leipzig, Germany, January 25-27 2012*, ACM International Conference Proceedings Series. ACM, 2012.

[9] M. Helvensteijn, R. Muschevici, and P.Y.H. Wong. Delta Modeling in Practice, a Fredhopper Case Study. In *Proceedings of the 6th International Workshop on Variability Modelling of Software-intensive Systems, Leipzig, Germany, January 25-27 2012*, ACM International Conference Proceedings Series. ACM, 2012.

[10] Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer, 1980.

[11] K. Pohl, G. Böckle, and F. Van Der Linden. *Software Product Line Engineering: Foundations, Principles, and Techniques*. Springer, Heidelberg, 2005.

[12] I. Schaefer. Variability Modelling for Model-Driven Development of Software Product Lines. In *Intl. Workshop on Variability Modelling of Software-intensive Systems (VaMoS 2010)*, 2010.

[13] I. Schaefer, L. Bettini, V. Bono, F. Damiani, and N. Tanzarella. Delta-oriented Programming of Software Product Lines. In *SPLC*, volume 6287 of *LNCS*, pages 77–91. Springer, 2010.

[14] I. Schaefer, A. Worret, and A. Poetzsch-Heffter. A Model-Based Framework for Automated Product Derivation. In *Proc. of Workshop in Model-based Approaches for Product Line Engineering (MAPLE 2009)*, 2009.